

The 3D Redeployment of Nodes in Wireless Sensor Networks with Real Testbed Prototyping

Sami Mnasri¹ , Adrien Van Den Bossche¹, Nejah Nasri², and Thierry Val¹

¹ University of Toulouse, UT2J, CNRS-IRIT-IRT, Toulouse, France
{Sami.Mnasri,Vandenbo,val}@irit.fr

² ENIS, LETI, University of Sfax, Sfax, Tunisia
nejah.nasri@isecs.rnu.tn

Abstract. In wireless sensor networks (WSNs), prototyping systems facilitate the realization of real node deployment, enabling to test new algorithms, protocols, and networking solutions. This paper investigates the 3D indoor redeployment problem in WSNs by finding the positions where nodes should be added in order to improve an initial deployment while optimizing different objectives. For this purpose, an approach based on a recent evolutionary optimization algorithm (NSGA-III) is used. The latter algorithm is hybridized with a strategy of incorporating of the user preferences (PI-EMO-VF). The major contributions of this work are as follows: testing the NSGA-III efficiency in the case of real world problems, comparing it with another recent many-objective algorithm (MOEA/DD), and incorporating the concept of preferences of users into NSGA-III. The real experiments performed on our testbeds indicate that the results given by the proposed algorithm are better than those given by other recent optimization algorithms such as MOEA/DD.

Keywords: Testbed · Prototyping · 3D indoor deployment · DL-IoT · NSGA-III · Preference · Many-objective optimization

1 The Problem of 3D Indoor Redeployment in WSNs

The redeployment of nodes in WSNs consists in identifying the number and positions of the nodes to be added to an initial configuration of nodes. It is a process that greatly influences the performance of a network. In this paper, we study the problem of redeploying nodes in three-dimensional spaces which represent the real topology of the RoI (region of interest) better than two-dimensional spaces. The improvement of the initial 3D indoor deployment is achieved by the addition of a set of nodes in a deterministic way while optimizing different objectives, such as localization, coverage, link quality, connectivity, or the lifetime of the network.

In fact, the recourse to use real experiments rather than simulations is due to the increasing simplicity of prototyping of communication devices and access to hardware. The major contribution of using real experiments than simulations is the realism of the obtained results. In this regard, the deployment of nodes with real prototyping that incorporates human experience is a more interesting to researchers than approaches

which are based on theoretical assumptions, simulations or formal calculations. Indeed, the objective of our work is to finely represent the real world using real physical nodes (36 TeensyWiNo nodes). These nodes are deployed using our rapid prototyping platform. Details of the implementation of this platform are presented in the experimental section.

In order to obtain the best distribution of the nodes in the RoI, while satisfying the mentioned objectives, we propose an approach which is based on an evolutionary genetic optimization algorithm, the NSGA-III [1]. Indeed, genetic algorithms (AGs), of which NSGA-III is a recent variant, are based on a set of individuals constituting the population. Each individual in this population has a genotype that encodes a candidate solution for a particular problem. After the determination of the initial population, a set of mechanisms such as recombination, mutation, and selection ensure the convergence of the population towards a better solution.

The rest of the paper is composed of the following sections: The new NSGA-III hybrid algorithm is detailed in Sect. 2. Then, our experiments on real testbeds are presented in Sect. 3. Finally, a conclusion and different perspectives are identified in Sect. 4.

2 A Many-Objective Hybrid Algorithm (PI-NSGAIII-VF) for the 3D Indoor Redeployment Problem

2.1 The NSGA-III Algorithm

NSGA-III is a new many-objective evolutionary optimization algorithm (MaOA) which is designed to solve many-objective problems (MaOPs). Based on the concept of reference points, NSGA-III is introduced as an extension of the NSGA-II algorithm [2] to resolve to problem of bad performance of the latter when the number of objectives exceeds three. As in the case of the MOEA/D algorithm [3], the NSGA-III is based on the concept of generating weight vectors to identify the reference points which are disseminated in the objective space.

In this regard, Algorithm 1 shows one generation of the NSGA-III algorithm. Indeed, for each generation of a solution, the values of the objective function are normalized to a binary value. Thereafter, each solution is associated with a reference point according to the perpendicular distance of each reference point to the reference line. This ensures having a uniform distribution of the reference points in the normalized hyper-plane.

Therefore, a hybrid population composed of the parent and an offspring is obtained. Subsequently, this population is divided into a set of non-domination levels using a non-dominated sorting procedure. The solutions constituting the first level serve as the next parents so on and so forth. Next, the solutions of the last acceptable level are selected using a niche preservation operator. Finally, a maximum number of iterations serves as the termination condition of the algorithm. In the case of many-objectives problems, NSGA-III has been shown to perform better than MOEA/D and NSGA-II for several theoretical test problems [1]. Remain the NSGA-III test on a real problem, compare it with other many-objective algorithms like MOEA/DD and incorporate it with a user preference procedure.

Algorithm 1 . One generation of the NSGA-III

Input: P_0 (Initial Population), N_{Pop} (size of population),
 t (iteration) = 0, It_{max} (Max iteration).
Output : P_t
While $t < It_{max}$ **do**
 Create Offspring Q_t
 Mutation and recombination on Q_t
 Set $R_t = P_t \cup Q_t$
 Apply non-dominated sorting on R_t and find F_1, F_2, \dots
 $S_t = \{\}, i = 1;$
 While $|S_t| \leq N_{Pop}$ **do**
 $S_t = S_t \cup F_i$
 $i = i + 1$
 End While
 If $|S_t| = N_{Pop}$ **do**
 $P_{t+1} = S_t$
 Else
 $P_{t+1} = \bigcup_{j=1}^{J-1} F_j$
 Normalize S_t using min and intercept points of each objective
 Associate each member of S_t to a reference point
 Choose $N_{Pop} - |P_{t+1}|$ members from F_l by niche preserving operator
 End if
 $t = t + 1;$
End While

2.2 Incorporation of Preferences

When solving a many-objective problem, MaOAs often provide a set of non-dominated solutions that are close as possible to the Pareto front (PF). In the case of real-world problems, which are known by a large number of objectives, the size of the population and the number of needed solutions exponentially depend on this number of objectives [4]. On the other hand, the user, often called the decision-maker (DM), usually only needs a very small number of non-dominated solutions. Thus, it is preferable to concentrate the search on a set of specific regions guided by the preferences of the user. Based on the integration time of the preferences, we can classify the preference procedures into three main classes [5]: A priori, interactive and a posteriori ones. The most relevant class is the second one, because it allows the progressive engagement of the preferences of the DM and the readjustment of its decisions as the algorithm generations evolve. Nevertheless, few research papers propose solutions for the interactive integration of the DM preferences. Our paper proposes a new hybridization scheme that incorporates NSGA-III into an interactive preference algorithm called PI-EMO-VF [6] where the DM is asked to interactively integrate its preferences in order to guide the searches towards a specific subset of the PF. Indeed, incorporated into any evolutionary multi-objective optimization (EMO) algorithm, PI-EMO-VF is a generic procedure that uses an approximate value function which is progressively generated. Indeed, after each small number of generations (iterations) of the EMO algorithm used (NSGA-III in our case), a set of non-dominated uniformly distributed solutions is determined, and the DM proposes its preferences (in general, an information about the relationship between the solutions). The DM can either classify all solutions from the best to the worst, or provide partial preference information. The given information helps to construct an increasing

polynomial value function which determines the stop condition. More details on the PI-EMO-VF procedure are given in [6].

3 Experiments with Real Prototyping

A testbed can be any platform that allows to test and experiment new network protocols, models, or technologies in order to prove its effectiveness in real-world environments. Indeed, formal analyzes and simulations cannot exactly reproduce the technical and physical characteristics of the real environments. Also, the current tendency is to experiment protocols and algorithms with real environments [7]. Therefore, the experiments carried out on our testbeds allow reducing the gap between the theory and the practice in the WSN deployment problems.

3.1 Parameters of Experiments

The following parameters were used in our experiments:

- Number of nodes: 36 (30 fixed nodes, 6 added ones)
- Antenna model: transceiver RFM22
- Distribution of nodes: $200 * 200 \text{ m}^2$
- Bit rate: 256 kbps
- Transmission power: 100 mW
- Indoor transmission range: 7 m
- Modem configuration: 12 # GFSK_Rb2Fd5
- Reception gain: 50 mA
- Indoor Sensing Range: 8 m
- Frequency: 434.79 MHz
- Modulation model: 125 Kbit/s GFSK
- Number of messages: 1000
- Average of runs: 20 experiments
- Message-wait: 5
- Tx power: 7 (which is the maximum of the standard RFM22)
- Length of the message: 16

3.2 Description of the Prototyping Platform

In our experiments, the deployed nodes are TeensyWiNo ones. They are based on the WiNoRF22 nodes, and equipped with several types of sensors (pressure, temperature, acceleration, brightness, etc.). These WiNo nodes have the advantage of allowing the developer to access to low layers. Indeed, this facilitates control of the average access time and the sleep one, the restricted memory and the CPU time. This control is necessary in order to ensure compliance with the real-time constraints and the management of drastic energy-saving policies. Moreover, these nodes with their effective material energy features (several months of operation using two AAA batteries), allow hosting

protocols with high temporal constraints. Figure 1 illustrates an example of the deployed WiNo nodes.

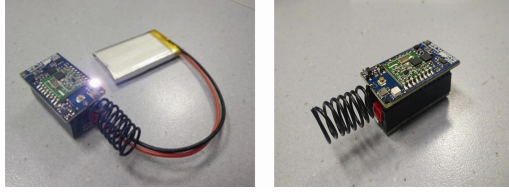


Fig. 1. The used Teensy WiNo nodes

The compatibility of these nodes with the Arduino environment enables researchers to easily integrate software bricks (processing algorithms, prototyping solutions, etc.) and hardware ones (sensors, actuators, interaction devices, etc.) which enables the feedback from users.

3.3 Results

Conduct of the Experiments. 30 fixed nodes are initially deployed in known positions that are chosen according to the application needs of the users. Six added nodes, having positions that are identified by the tested optimization algorithms. The experimental scenario is as follows: Initially, all nodes are flashed. Afterwards, they are sent the initial configuration parameters (transmission power, etc.). Then a randomly selected node sends a message to the other nodes of the network and records the RSSI (received signal strength indicator) and FER (frame error rate) measurements detected by these nodes and the received measurements (RSSI and FER) of the nodes receiving the signal. After a predefined waiting time, the sender ends the process. Afterwards, the transmitter is changed in such a way that each time a receiver node becomes a transmitter. These steps are repeated until 36 experiments (one send and 35 receptions in each experiment) are performed. The obtained values representing the relations between all the nodes are collected and recorded in two connectivity matrices (an RSSI matrix and an FER one). These two matrices give also information about the neighbors of each node. Indeed, we consider a node i as a neighbor to another node j if the following two conditions are satisfied: (a) The average of the transmitted RSSI between the two nodes (from i to j and vice versa) is higher than a predefined threshold (set to 100 in our experiments); and (b): The average FER value is less than a predefined threshold (set to 1/10 in our experiments).

To assess the impact of the new chosen positions of the added nodes on the performance of the network, this process is repeated several times (20 times in our experiments, given the variation of RSSI and FER rates and the stochastic nature of the evolutionary algorithms). The performance of the proposed algorithm (PI-NSGAI-III-VF) is compared to the MOEA/DD [8] which is another recent competitive MaOA.

Evaluation of the RSSI Rates. The evaluation of a set of the considered objectives in our study (such as localization, link quality, and connectivity) is achieved using the RSSI rate measurement. Indeed, in our experiments, the localization is based on a protocol that combines the RSSI rates and the Distance-VectorHop protocol. Therefore, the localization is proportionally dependent on the RSSI rate. Figure 2 illustrates the average RSSI rate between each node and the other ones. The RSSI is represented by a value between 0 and 256, which is convertible to a dBm measure.

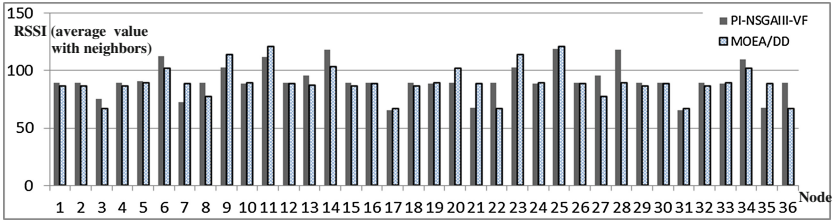


Fig. 2. Average rates of RSSI between nodes

Evaluation of FER Rates. Similarly, the measurement of the FER rate is used to assess the coverage and the quality of link between nodes. Indeed, the coverage is proportionally dependent on the FER rate. Figure 3 illustrates the average rate of FER between each node and the other ones.

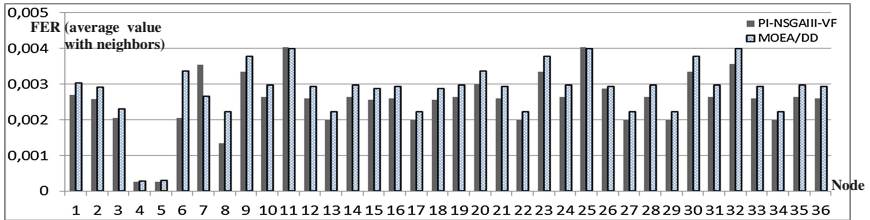


Fig. 3. Average rates of FER between nodes

4 Conclusion and Perspectives

This paper aims to resolve the problem of 3D indoor redeployment in WSNs. It presents a real world deployment which is based on a real prototyping with WiNo nodes. To find the best node positions, we use a hybrid algorithm, called PI-NSGA-III-VF, which integrates the evolutionary optimization algorithm NSGA-III into the PI-EMO-VF preference procedure. This allows testing the performance of the NSGA-III with a real world problem, and evaluating its behavior by hybridizing it with the PI-EMO-VF. Different extensions of this work can be envisaged. Among others, supporting different transmission protocols by developing them on the OpenWiNo library which lacks the implementation of several standard protocols. Moreover, considering several other constraints

that makes experiments more realistic, such as the mobility of the deployed nodes. Also, taking into account the existence of obstacles in the deployment area.

References

1. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2014). doi:[10.1109/TEVC.2013.2281535](https://doi.org/10.1109/TEVC.2013.2281535)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). doi:[10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
3. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007). doi:[10.1109/TEVC.2007.892759](https://doi.org/10.1109/TEVC.2007.892759)
4. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Adaptation of scalarizing functions in MOEA/D: an adaptive scalarizing function-based multiobjective evolutionary algorithm. In: Ehrgott, M., Fonseca, Carlos M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) *EMO 2009*. LNCS, vol. 5467, pp. 438–452. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-01020-0_35](https://doi.org/10.1007/978-3-642-01020-0_35)
5. Jaimes, A.L., Montañó, A.A., Coello, C.A.C.: Preference incorporation to solve many-objective airfoil design problems. In: *IEEE Congress of Evolutionary Computation (CEC2011)*, pp. 1605–1612. New Orleans, LA (2011). doi:[10.1109/CEC.2011.5949807](https://doi.org/10.1109/CEC.2011.5949807)
6. Deb, K., Sinha, A., Korhonen, P., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. *IEEE Trans. Evol. Comput.* **14**(5), 723–739 (2010). doi:[10.1109/TEVC.2010.2064323](https://doi.org/10.1109/TEVC.2010.2064323)
7. Mnasri, S., Nasri, N., Val, T.: An overview of the deployment paradigms in the wireless sensor networks. In: *Proceedings International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2014)*, Tunisie, 04–07 November 2014
8. Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **19**(5), 694–716 (2015). doi:[10.1109/TEVC.2014.2373386](https://doi.org/10.1109/TEVC.2014.2373386)

Ad-hoc, Mobile, and Wireless Networks

16th International Conference on Ad Hoc Networks and
Wireless, ADHOC-NOW 2017, Messina, Italy, September
20-22, 2017, Proceedings

Puliafito, A.; Bruneo, D.; Distefano, S.; Longo, F. (Eds.)

2017, XIV, 396 p. 163 illus., Softcover

ISBN: 978-3-319-67909-9